

Can LLMs Keep Up with Library Changes?

An Exploratory Study on LLM-Generated Code

Xiangrong Lin

The State Key Laboratory of
Blockchain and Data Security,
Zhejiang University
Hangzhou, Zhejiang, China
22321084@zju.edu.cn

Jiakun Liu

Harbin Institute of Technology
Harbin, China
jiakunliu@hit.edu.cn

Lingfeng Bao*,

The State Key Laboratory of
Blockchain and Data Security,
Zhejiang University
Hangzhou, Zhejiang, China
lingfengbao@zju.edu.cn

SANAR 2026

学生番号 : 60276216

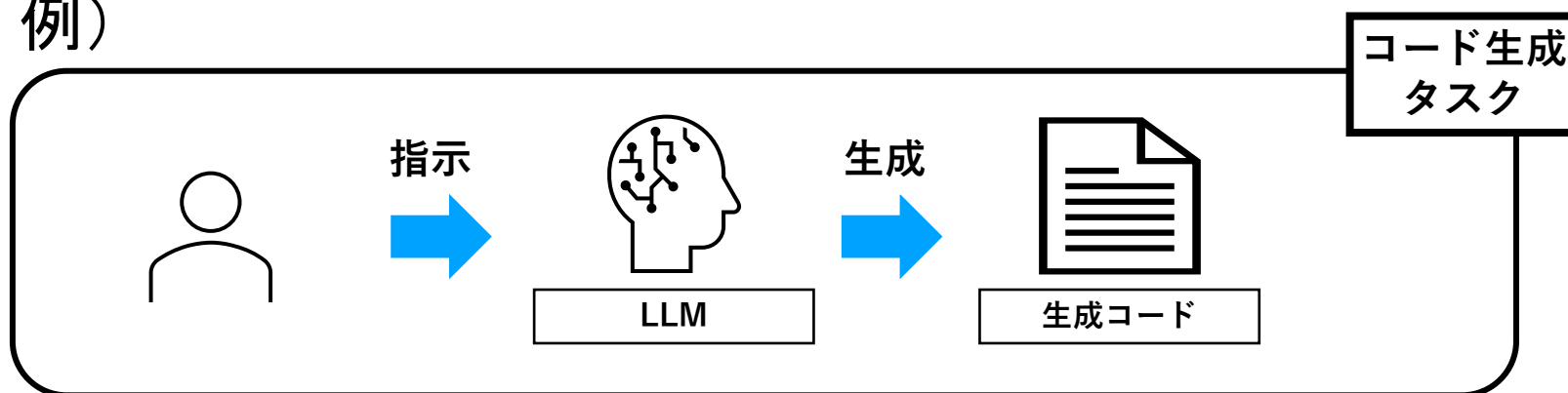
発表者: 福井 遥陽

背景

近年のLLM

LLMは過去のデータでトレーニングされているため、トレーニング時に非推奨のライブラリを含む場合がある。

例)



LLMが生成するコードに不適切なライブラリが含まれる可能性

目的

現状の課題点

継続的に更新・変更されるライブラリをchatGPTといったLLMがどう扱うかが不明である。

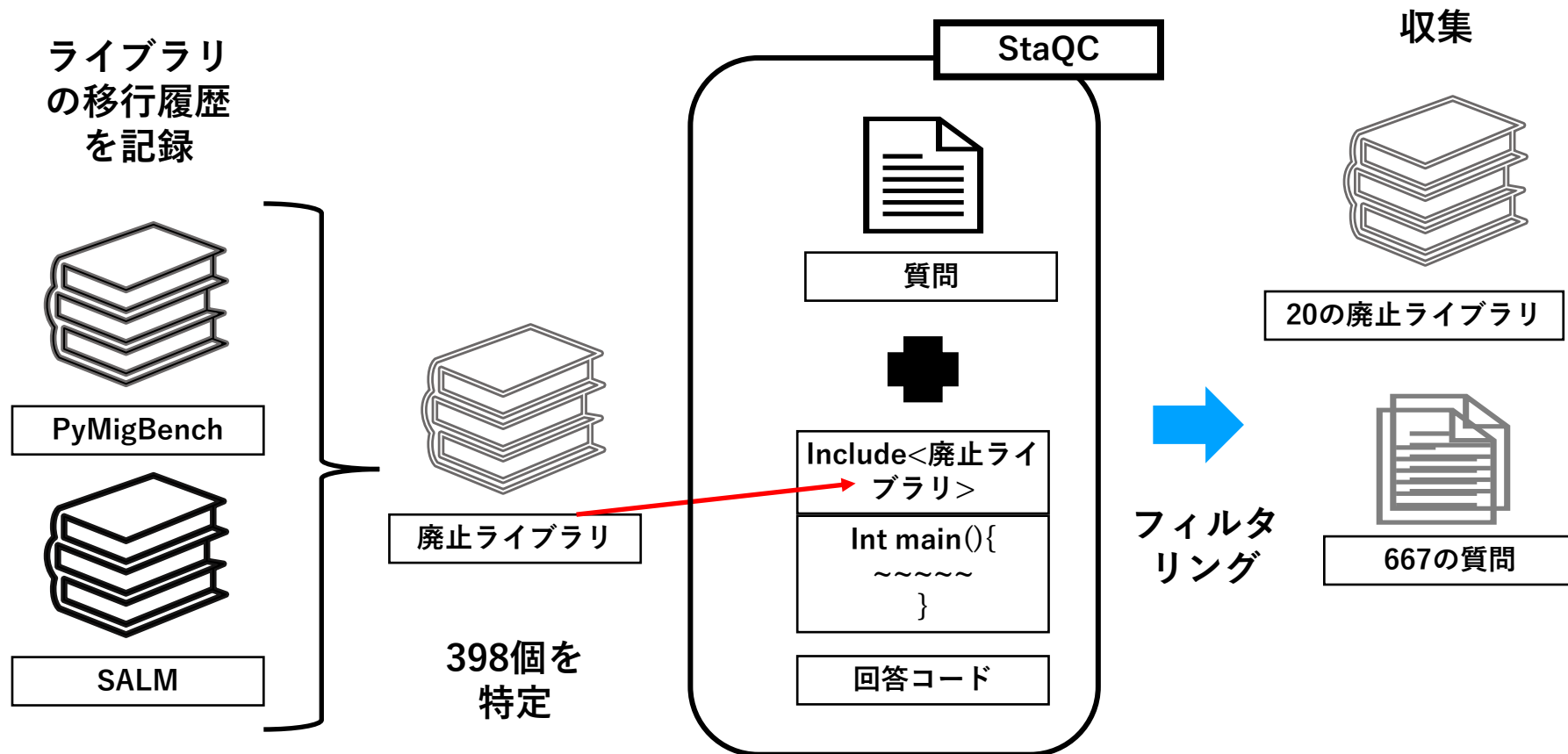
最新のライブラリを生成するのか？
古いライブラリも生成する可能性は？

本研究の目的

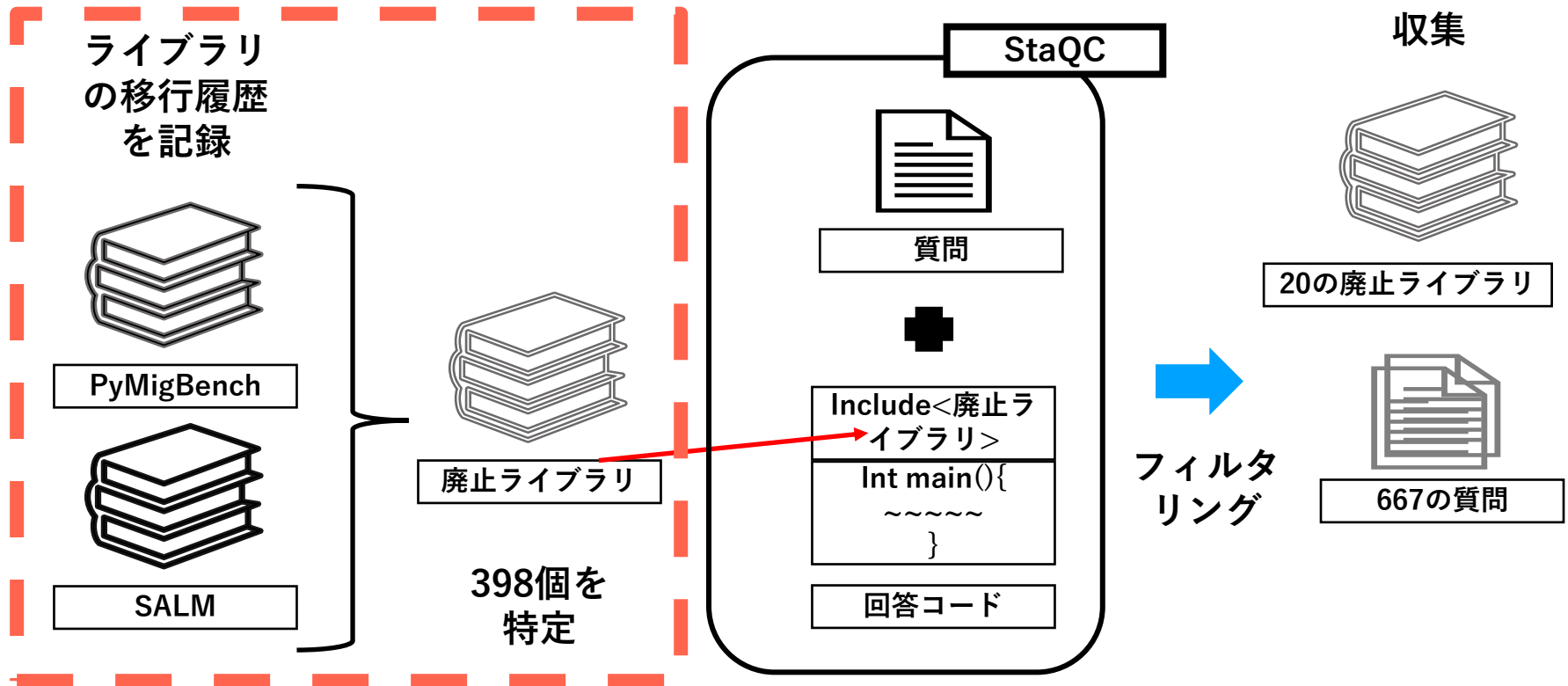
既存のLLMが、廃止されたライブラリに関連するタスクでどれ程情報をアップデートできているかを評価

実験

ワークフロー

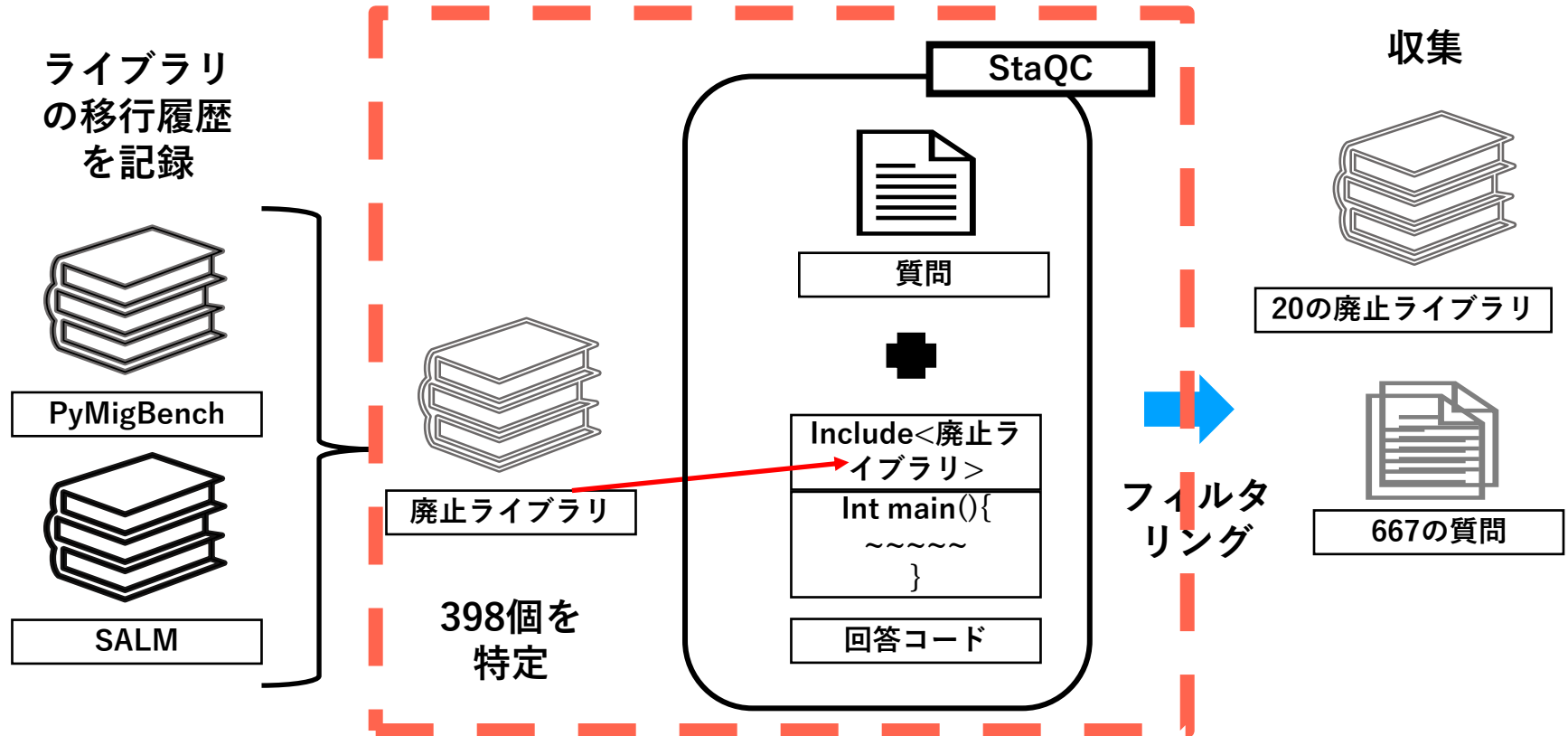


実験



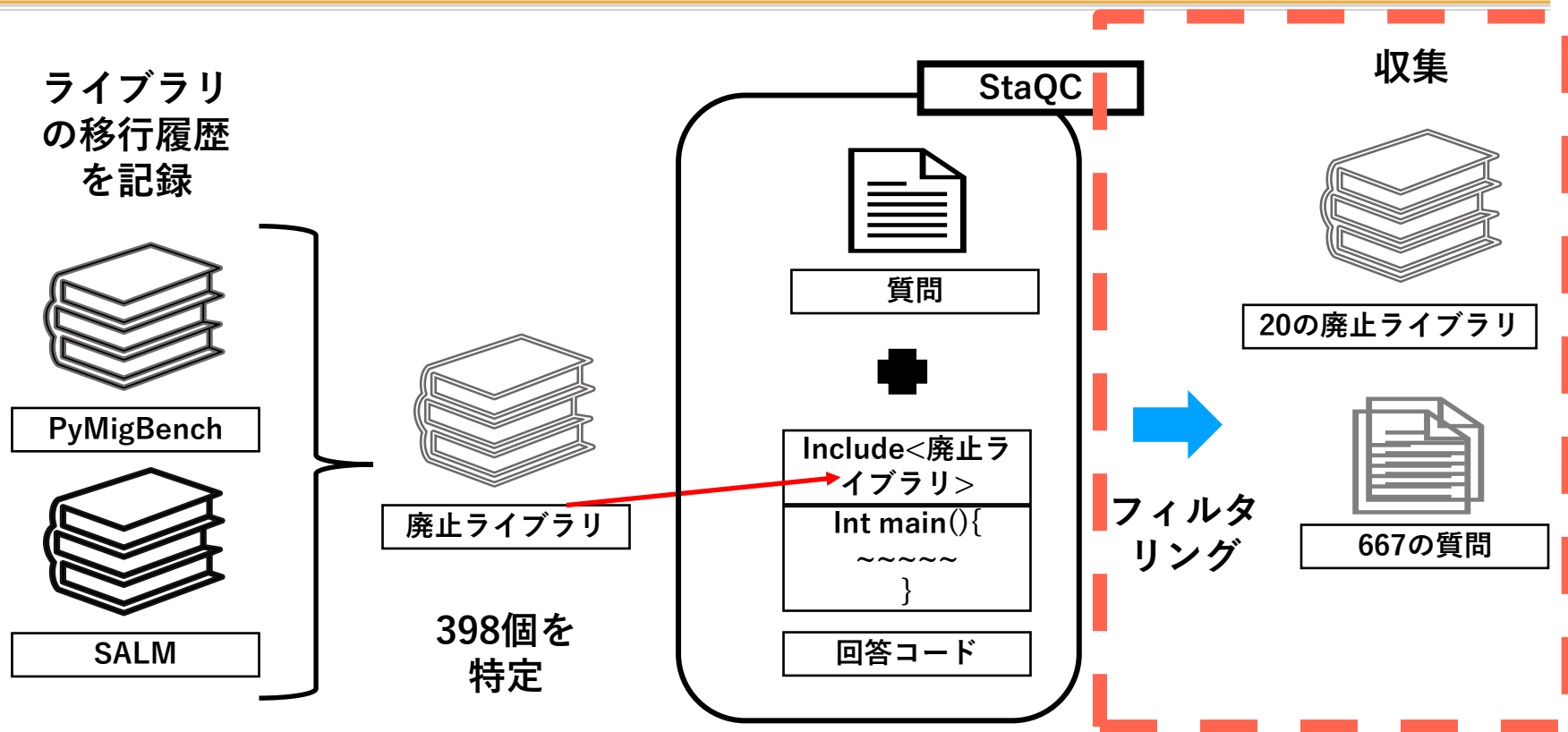
1. データセットの収集：pythonのライブラリ移行ベンチマークの「PyMigBench」「SALM」から、398の廃止されたライブラリを特定

実験



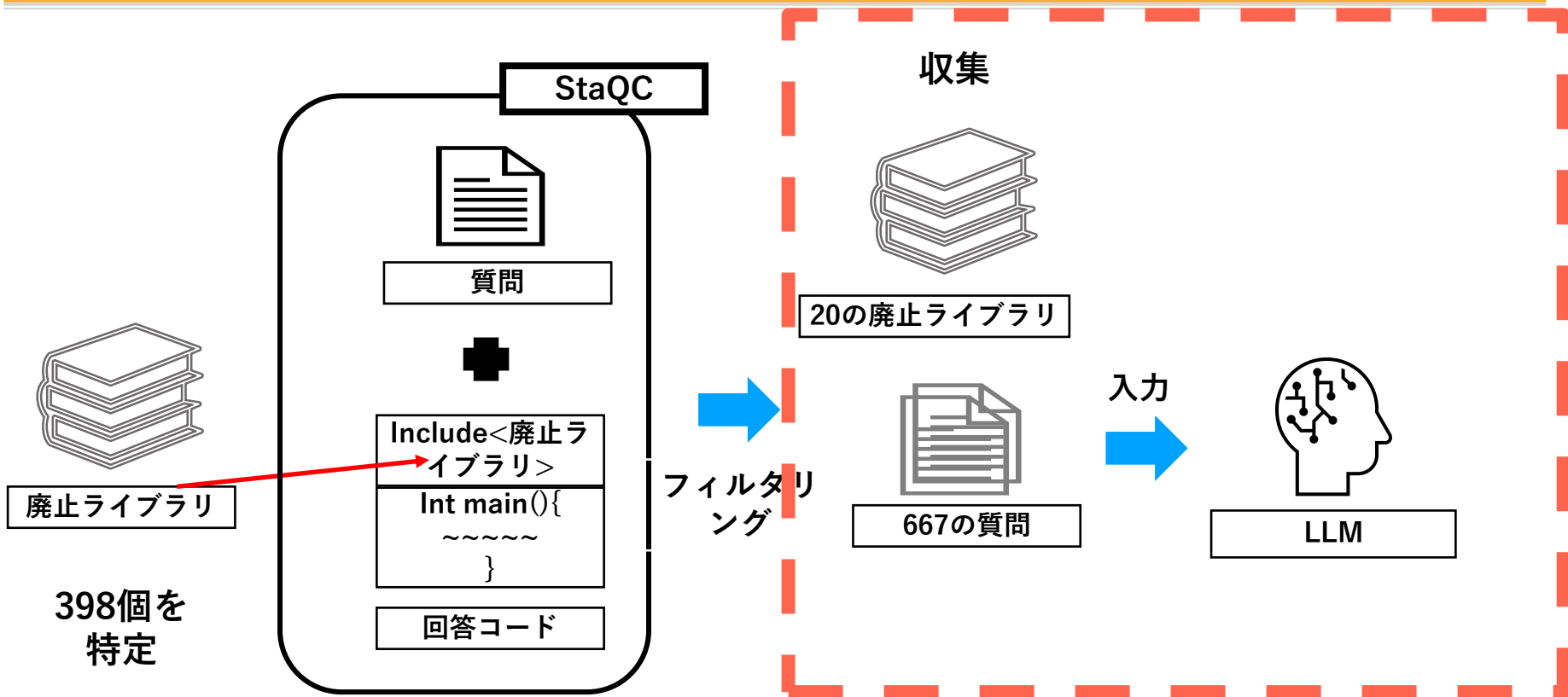
2. Stack Overflow質問の選定：データセットから，これらの廃止済みライブラリに関連する「やり方」形式の質問を抽出

実験



3. フィルタリングと分類：プロジェクト固有の特殊な移行を除外し、最終的に20の廃止ライブラリと667の質問に絞り込む

実験



4. LLMによるコード生成: 5つの主要なLLM (Qwen-Plus, LLaMA3.3, DeepSeek-V3, GPT-3.5-Turbo, GPT-4o) を使用し, 標準化されたプロンプトでコードを生成

実験結果

RQ1：廃止要因とドメインの影響

廃止ライブラリ	移行先	GPT-4o(LRP)	Qwen(LRP)
BeautifulSoup	Bs4	97.96%	100%
Optparse	Argparse	82.35%	100%
Pickle	Json,dill	36.59%	33.33%
Pytz	Zoneinfo	88.89%	83.33%
twisted	asyncio	19.57%	4.26%

- 長期メンテのライブラリ：精度が約90%に
- セキュリティ主導：精度が約37%に低下
- ドメイン別：システムユーティリティでは90%，データ処理ドメインでは20%に低下

実験結果

RQ2：質問文の説明の影響

指標	GPT-4o（機能主導）	GPT-4o（名前明示）
精度(LRP)	85.28%	65.93%
関連性(LSR)	84.91%	93.00%

指標	Qwen（機能主導）	Qwen（名前明示）
精度(LRP)	82.35%	48.09%
関連性(LSR)	84.20%	97.14%

※LSR:LLMが（新旧問わず）そのタスクに関連する適切なライブラリを選択できたかの割合

※LRP:関連ライブラリを選んだうち、どれだけ「最新の（移行後の）ライブラリ」を選べたかの割合

実験結果

RQ3 : プロンプト改良の影響

対象	改良無し(LRP)	非推奨を明示	代替案を提示
twisted（全モデル平均）	13%	80%	98%
pytz（全モデル平均）	9%	87%	100%

- 非推奨の明示 : 「このライブラリは古い」と伝える
- 代替案の推奨 : 「最新の代替案はこれである」と伝える

考察

データ処理ドメインでの精度低下

- ライブラリの移行を行う場合、扱うデータ形式（.xlsから.xlsx）の変化を伴うため、十分に対応できていない

名前明示型での低下理由

- ユーザー指示にライブラリ名がある場合、非推奨であってもそのライブラリを使用する傾向にある

まとめ

結論

- LLMが生成したコードのライブラリ選択, 特にセキュリティが絡む箇所は, 開発者が慎重にレビューする必要
- プロンプトを作成する際は, 特定のライブラリ名を指定するよりも, 「実現したい機能」を記述するほうが、より現代的な回答を得られる.